REMARKS

Claims 1-4 and 16-48 are pending in the present patent application. Applicant amends Claim 1. Applicant adds Claims 49-65. Consideration and examination of pending Claims 1-4, 16-65 is respectfully requested.

Amendment to Claim 1

Applicant amends Claim 1 to provide proper antecedent basis.

Rejection of Claims 1, 2, 17-20, 24, 30, 36, 37 and 41-48 under 35 U.S.C. § 103(a)

The Examiner rejects Claims 1, 2, 17-20, 24, 30, 36, 37 and 41-48 under 35 U.S.C. § 103(a) as being unpatentable over Richek et. al in view of Nuttall et al. Specifically, the Examiner states:

> "Richek taught a configuration system, as example in claim 1, including: defining in the computer system an element model (system configuration file) consisting of [sic] elements used to configure elements of the model (e.g., see col. 21, lines 21-et al) and creating a plurality of components of the system in response to configuration requests (e.g., see col. 22, lines 10-et seq)."

The Examiner admits that Richek does not teach creating components as instances of elements in the model. Applicant agrees with the Examiner. However, the Examiner states that Nuttall teaches modeling a physical system of elements and creating components of a system as instances of elements (e.g., col. 5, lines 61-et seq) in response to configuration requests. The Examiner states that it would have been obvious to combine the teachings of Richek and Nuttall especially using the object oriented

information model system of configuration of Nuttall's system in Richek's system will make the system a flexible system.

Applicant respectfully disagrees. Applicant submits that neither Richek nor Nuttall either alone or in combination teach, suggest or disclose the claimed invention. Applicant submits that, for at least the following reasons, the claimed invention is allowable over Richek and Nuttall either alone or in combination:

1.   Richek does not teach, suggest or describe creating a plurality of components of a system in response to a configuration request;

2.   Richek does not teach, suggest or describe an element model having elements where components of the system are instances of one or more elements of the element model;

3.   Nuttall does not teach, suggest or describe creating components as instances of elements in the model;

The following provides a discussion of these reasons.

1.   <u>Richek does not teach, suggest or describe creating a plurality of components of a system in response to a configuration request.</u>

The Examiner cites Richek at col. 22, lines 10-et al. in support of a notion that Richek teaches creating a plurality of components of the system in response to a configuration request. Applicant respectfully disagrees.

Richek does not create a plurality of components of a system in response to a configuration request. Richek identifies settings (i.e., address locations, DMA channels, interrupt lines, etc.) for a computer system's circuit

boards. Settings identification is in fact illustrated in the portion of Richek cited by the Examiner (i.e., col. 22, lines 10-et al.). In the cited portion, Richek describes the three functions that are performed in Richek to identify the settings for circuit boards for use in allocating system resources and initializing the circuit boards. According to Richek at col. 22, lines 10-et al., the three functions are: determining the available settings, choosing a set of settings, and storing the settings (referred to in Richek as the configuration information). As it can be seen from Richek, these three functions merely identify a circuit board's settings. They do not create components of a system in response to a configuration request.

Figures 4, 10A-10B, 11 and 12 further support the position that Richek describes a process of identifying initialization settings for circuit boards. Figure 4 specifies three routines that are used in Richek (see steps 500 and 502). Richek states (beginning at col. 37, line 35), that the three routines, the Process, Allocate and Backtrack routines, are performed in Figures 10A-10B, 11 and 12, respectively.

In Figures 10A-10B, Richek processes an array that contains the settings options for each circuit board. Figure 11 of Richek determines whether a conflict exists with another circuit board's settings, and Figure 12 determines whether it is possible to backtrack to eliminate a conflict if one exists. That is, in the Process subroutine (Figures 10A-10B), the Allocate subroutine (Figure 11) is called for each circuit board in the array to allocate the resources for the circuit board. If a resource allocation conflict is detected in the Allocate routine, an attempt is made to resolve the conflict (e.g., by attempting to share

the resources or using alternate resources) with respect to the current circuit board. If the conflict resolution is unsuccessful, the Backtracking routine (Figure 12) is called to resolve the conflict by allocating alternate resources to the circuit boards that are a party to the conflict. If a conflict cannot be resolved by either sharing resources or allocating alternate resources, an error message is displayed (see Figure 4, steps 504 and 505).

As described in the portion of Richek cited by the Examiner and illustrated in Figures 10A-10B, 11, and 12, Richek merely describes a process of identifying settings for circuit boards. Richek does not teach creating a component of a system in response to a configuration request.

Application agrees with the Examiner statement that Richek does not teach creating components as instances of elements in the model. Applicant submits, that Richek fails to teach, suggest or describe in any respect creating a plurality of components of a system in response to a configuration request.

2.   <u>Richek does not teach, suggest or describe an element model having elements where components of the system are instances of one or more elements of the element model.</u>

The Examiner states that Richek teaches "defining in the computer system an element model (system configuration file) consisting of [sic] elements used to configure elements in the model." The Examiner cites col. 21, lines 21-et al. in support of this position. Applicant respectfully disagrees. Applicant submits that the Examiner's characterization of the claim language is inaccurate. Applicant submits that Richek does not teach an element model as in Claim 1 consisting of elements used to configure a system where

components of the system are instances of one or more elements of the element model.

As stated in Richek (at col. 4, lines 47-50):

"[t]he information in the configuration file consists of a series of parameters which serve two general purposes: common computer system resource allocation and circuit board initialization.

Thus, in Richek, a configuration file is merely a collection of allocation and initialization parameters. This is illustrated in Richek, lines 21-27 of col. 21 (cited by the Examiner) which provide an example of text of an configuration file that identifies two different choices of baud rates and settings for a serial port. That is, the configuration file specifies parameters that identify port setting alternatives for a specific circuit board.

Richek does not teach, suggest or describe an element model comprising elements instances of which are used to generate a system. Richek does not teach, suggest or describe the claimed element model. Further, Richek does not create components of a system which are instances of elements of the model. Richek doesn't create. Richek merely identifies the resource and initialization parameters for circuit boards in a computer system.

3.   Nuttall does not teach, suggest or describe creating components as instances of elements in the model.

The Examiner cites col. 5, lines 61-et seq. for the proposition that Nuttall teaches modeling a physical system of elements and creating

components of a system as instances of elements. Applicant respectfully disagrees.

The portion of Nuttall that is cited by the Examiner appears to be cited because it contains the word instance. However, the word instance is used in Nuttall to describe an instance of an object-oriented object. Specifically, Nuttall defines an instance (at col. 6, lines 1-3) as:

> "a specific occurrence of an object, in which the object's attributes are populated with data."

Thus, in Nuttall, an instance is an object-oriented object that contains data.

Nuttall describes using an object-oriented approach to managing the data associated with a physical system. In other words, Nuttall merely describes defining a physical system as object-oriented objects that have attributes and behavior. As stated in Nuttall (beginning at col. 1, line 66), Nuttall's invention is an object-oriented information model of a physical system" where "physical pieces of equipment are represented as objects." Referring to Figure 1, Nuttall manages a physical system's data using a relational database data model (element 10), processes (elements 22, 24, 18, etc.) for operating on the data via object-oriented objects and means for exporting (external database generators 14a-n) and importing data (data port 26) to/from other application programs.

At col. 5, lines 60-67, Nuttall proposes using object-oriented analysis and design to define the attributes and behavior for object-oriented objects that correspond with specific elements of the system. At col. 6, beginning at

line 1 in Nuttall, the object-oriented objects are then populated with the data associated with an element of the physical system. As indicated above, Nuttall refers to an object-oriented object that is so populated with data as an instance.

Thus, in Nuttall, a data model is used to retain data that has been identified through object-oriented analysis for each physical piece of an existing, physical system. The data in the model is used to populate the attributes of instances of object-oriented objects.

Nuttall populates object-oriented objects with data that is stored in a relational database. Nuttall does not describe a method of generating a system configuration comprising defining a model and creating a plurality of components of said system configuration that are instances of one or more elements of the model. Nuttall does not teach, suggest or describe creating components of a system configuration where the components in the configuration are instances of one or more elements of the model in response to a configuration request. Further, Nuttall does not even teach, suggest or describe a configuration request or responding to a configuration request by creating a plurality of components instance of one of the instances in the model.

New Claims

In addition to Claims 49-65, Applicant has added new Claims 49-65. Applicant submits that new Claims 49-65 are fully supported by the

specification, claims, and figures as originally filed, and that no new matter is added.

Claims 3, 4, 16, 21-23, 25-29, 31-33 and 38-40 are rewritten as Claims 49-65 in independent form including all of the limitations of the base claim and any intervening claims. Applicant submits that Claims 49-65 are in condition for allowance.

Conclusion

For the foregoing reasons, Applicant contends that none of the references cited, either alone or in combination, teach, describe, or suggest the present invention. Applicant contends that pending Claims 1-4, 16-65 are in condition for allowance.

Respectfully submitted,

Hecker & Harriman

Date: 4/23/99

By: _____
J. D. Harriman II
Reg. No. 31,967

1925 Century Park East
Suite 2300
Los Angeles, CA 90067
(310) 286-0377
File No.: 85160.911CII

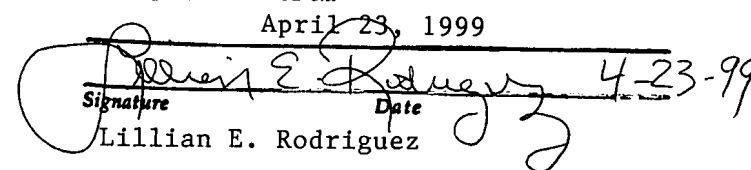85160.911CII   33